1

DLA-91-P81109

THE REVIC ADVISOR (REVAD): AN EXPERT SYSTEM PREPROCESSOR TO A PARAMETRIC SOFTWARE COST ESTIMATING MODEL

September 1991

OPERATIONS RESEARCH AND ECONOMIC ANALYSIS OFFICE

DEPARTMENT OF DEFENSE
DEFENSE LOGISTICS AGENCY

September 1991 Final

The REVIC Advisor (REVAD): An Expert System
Preprocessor to a Parmetric Software Cost Estimating
Model

Alan R. Greve, Richard M. Silla, Keith D. Ernst,
Dennis L. Skoneczka

Defense Logisitics Agency
Operations Research & Economic Analysis Office (DLA-LO)
Cameron Station
ALexandria, VA 22304-6100

DLA-91-P81109

Defense Logistics Agency (DLA-EE)
Cameron Stai-on
ALexandria, VA 22304-6100

Public Release; Unlimited Distribution

Defense Contract Management Command (DCMC) engineers perform a Technical Analysis of a Cost Proposal (TACP) whenever a contractor submits a proposal in an environment of little or no competition. This environment applies to most weapons systems software proposals. The Department of Defense purchased an estimated $34 billion of software embedded in weapons systems in fiscal year 1990. Tools are needed to assist DCMC engineers in performing TACPs involving software. REVAD, coupled with REVIC, provides a flexible analytical framework from which the engineer may follow the analysis wherever it may lead. In addition to expert decision rules and an online software glossary, REVAD produces a complete report of the software cost estimating session, which the engineer may include in the TACP report. Based on the favorable results of a field test and the potential of huge savings from more effective TACPs involving software, DCMC should distribute REVAD throughout the Command as a tool for software estimating.

68

Cost Estimating, Contracting, Cost Proposal, Expert System

UNCLASSIFIED UNCLASSIFIED UNCLASSIFIED

DLA-91-P81109

THE REVIC ADVISOR (REVAD): AN EXPERT SYSTEM PREPROCESSOR TO A PARAMETRIC SOFTWARE COST ESTIMATING MODEL

September 1991

Alan R. Greve and Richard M. Silla
Operations Research and Economic Analysis Office

Keith D. Ernst
Defense Contract Management Area Office,
Twin Cities

Dennis L. Skoneczka
DLA Systems Engineering Division

DEPARTMENT OF DEFENSE
DEFENSE LOGISTICS AGENCY
OPERATIONS RESEARCH AND ECONOMIC ANALYSIS OFFICE
CAMERON STATION
ALEXANDRIA, VA 22304-6100

DEFENSE LOGISTICS AGENCY
HEADQUARTERS
CAMERON STATION
ALEXANDRIA, VIRGINIA 22304-6100

FOREWORD

The Defense Logistics Agency (DLA) Operations Research and Economic Analysis Management Support Office, Chicago, DLA DORO-C, has developed an expert system preprocessor to a parametric software cost estimating model (REVIC or Revised Intermediate COCOMO) in response to a request from the Directorate of Program and Technical Support. This report documents the development of the preprocessor, titled REVAD (REVIC Advisor).

Defense Contract Management Command (DCMC) engineers perform a Technical Analysis of a Cost Proposal (TACP) whenever a contractor submits a proposal in an environment of little or no competition. This environment applies to most weapons systems software proposals. The Department of Defense purchased an estimated $34 billion of software embedded in weapons systems in Fiscal Year 1990.

Tools are needed to assist DCMC engineers in performing TACPs involving software. REVAD, coupled with REVIC, provides a flexible analytical framework from which the engineer may follow the analysis wherever it may lead. In addition to expert decision rules and an online software glossary, REVAD produces a complete report of the software cost estimating session, which the engineer may include in the TACP report.

Based on the favorable results of a field test and the potential of huge savings from more effective TACPs involving software, DCMC should distribute REVAD throughout the command as a tool for software cost estimating.

ROGER C. ROY
Assistant Director
Policy and Plans

CONTENTS

I. INTRODUCTION

A. Project Background. The Defense Logistics Agency Directorate of Program and Technical Support (DLA-E) requested that this expert system tool be developed.

Engineers of the Defense Contract Management Command (DCMC) perform a Technical Analysis of a Cost Proposal (TACP) at the request of the Administrative Contracting Officer (ACO) or price analyst. A TACP is usually requested when a contractor's proposal is made in an environment of little or no competition. This environment applies to most proposals for weapons systems software. In fiscal year 1990, DoD procured approximately $34 billion of software embedded in weapon systems.

DCMC needs improved software cost estimating tools to use on these software TACPs. One set of tools for software proposals involves the use of parametric cost models. Parametric modeling uses parameters or characteristics of a product to develop cost estimating relationships. Using these cost estimating relationships, cost estimates can be made by inputting values for critical parameters. For example, to estimate house building costs an analyst could input values for the number of square feet, the primary type of building material and the number of floors into a formula that describes these cost estimating relationships.

Several public domain parametric software cost estimating models have circulated throughout the country informally. Due to differing basic assumptions, most produced conflicting and often confusing results. The study entitled "Feasibility of Applying Expert System Technology to the Technical Analysis of a Cost Proposal" outlined the benefits of designing an Expert System front end for a recognized software cost estimating model for use by DCMC engineers.

REVIC (Revised Intermediate COCOMO) was chosen as the software cost model for several reasons. REVIC is based on Constructive Cost Model (COCOMO) constructs as outlined by Dr. Barry Boehm in his book Software Engineering Economics. COCOMO is an established, accepted technology in the software cost estimating field. REVIC is being maintained and updated in association with an active users group. REVIC, though copyrighted, has been donated for use by DoD and DoD contractors.

B. **Purpose.** REVAD (REVIC Advisor) provides an expert system front end to a parametric software cost estimating model (REVIC). REVAD assists engineers in the analysis of software TACPs in several ways. REVAD acquaints engineers with the parameters and terminology unique to software engineering. It guides the engineer through the selection of the most appropriate input values for the 23 parameters REVIC uses. These input values can then run REVIC directly without being re-input. A written report of the entire REVAD session is available for inclusion in the TACP report.

C. **Scope.** REVAD is **not** an automated TACP. When used in conjunction with REVIC, it provides the engineer with an intelligent framework of analysis for the TACP process. The fact finding required to adequately use REVAD should spawn new avenues of inquiry.

II. METHODOLOGY

An expert system approach was used. Mr. Keith Ernst, a DCMC engineer stationed at Defense Contract Management Area Office, Twin Cities, served as the parametric software cost model expert. His expertise and advice based on extensive software cost experience proved vital. Rules of thumb outlined by Dr. Barry Boehm in his book, Software Engineering Economics, as well as those incorporated in REVIC were also included in the expert system preprocessor. A field test of the REVAD prototype was conducted at the offices listed in section IV. The comments from the test respondents were also incorporated in the final model.

REVAD was built originally in the Teknowledge expert system shell titled M1. Due to memory constraints and other M1 limitations, REVAD was rewritten in VP Expert. REVAD stores its report output as an ASCII file that may be edited by most any word processing program. All parameter choices are saved in a file that can be directly accessed by REVIC to do the actual cost estimating. This saves the user from entering the same data twice. REVAD sessions may be interrupted at any point, the data saved and the session resumed later at the user's convenience. A complete user's guide for both REVAD and REVIC is attached.

System requirements include a Zenith 248 (or compatible) computer with 640K of main memory, a hard disk, a 5.25 inch floppy disk drive and a printer.

III. BENEFITS

REVAD coupled with REVIC provides a number of benefits:

- Increases dollar savings by providing a consistent, repeatable, analytical framework for software TACPs.

- Provides flexible starting point from which the engineer may follow the analysis wherever it leads.

- Aids novices in software engineering with shared expertise from recognized experts in the field.

- Provides common ground for negotiation with contractors by using COCOMO, a recognized software cost estimating technology.

- Enhances utility of TACP report with a complete written record of both the REVAD and REVIC sessions.

- Trains new engineers with software glossary and numerous explanatory screens.

- Uses an affordable software cost estimating tool, REVIC, that is being maintained and updated.

IV. CONCLUSIONS AND RECOMMENDATIONS

REVAD was distributed for field testing at DPRO Loral in Akron, Ohio, DPRO Harris in Palm Bay, Florida, and the Aerojet Residency in Azusa, California. Comments were also received from DPRO Northrop in Rolling Meadows, Illinois and the REVIC User's Group. The consensus of all was that, with refinements, REVAD coupled with REVIC is a valuable tool for use by DCMC engineers. Based on the results of this field test the DLA Program and Technical Support Directorate:

- should adopt REVAD coupled with REVIC and implement DCMC wide.

- should promote efforts to maintain and update REVAD and REVIC.

REVAD USER'S GUIDE
TABLE OF CONTENTS

INTRODUCTION TO REVAD
SECTION I
TABLE OF CONTENTS

TITLE PAGE

I. INTRODUCTION. Dr. Barry Boehm's COnstructive COst MOdel (COCOMO) methodology, outlined in his book, Software Engineering Economics, is a recognized standard for building software cost models in the software engineering field today. One model now available is called Revised Intermediate COCOMO (REVIC). The REVIC Advisor (REVAD) is an Expert System (ES) that assists the user in selecting REVIC input parameters. This section will provide brief background information and a discussion of main REVAD features. Note: **Parametric modeling of software costs is NOT meant to be the end of the TACP, but rather an important part of the process. It is expected that the fact finding required to adequately use the model will spawn new avenues of inquiry for the TACP process.**

A. Background. REVIC was developed by Major Ray Kile, an Air Force Reservist. As a civilian, he is a software engineer for Hughes Corp. REVIC is copyrighted, but donated for use by U.S. Government and industry. The results of the model, properly used and interpreted, should support most negotiations.

The model's outputs include the man months required for software development in six segments or phases. These six are: system engineering; preliminary design; critical design; code and unit testing; integration and test; development test and evaluation. All but the first and last of these are calculated using the results of the REVIC effort equations. Input parameters tailor the model to a specific proposal. Experience with the model may lead to contractor-specific adjustments to the underlying effort and schedule equations used in the model.

REVIC uses 18 different parameters and 5 additional project statistics. The parameters are broken down into four groups: product, computer, personnel, and project attributes. Each parameter is given a rating, generally from Very Low (VL) to Very High (VH). Each rating translates into a numerical value. Multiplying all 18 parameter value generates an environmental factor. This factor is multiplied by the nominal man months to provide a project-specific and contractor-specific estimate.

B. Tips for Using REVIC and REVAD. **Adequate preparation helps get the most out of any cost estimating technique. The engineer must read the relevant portions of the proposal and be familiar with the work done by the contractor in the past. This knowledge includes: types of projects worked on; cost/price history of those projects; types of cost estimating techniques used by the contractor on those projects; and a thorough explanation of how the contractor arrived at the costs for the** current proposal. Some of the information needed to run REVAD is listed on the form in Section 3, titled "REVAD DATA COLLECTION FORM".

In looking at any proposal, the engineer should be aware of software that may be hidden. Many components produced for the military today have software routines burned into a programmable read only memory (PROM) chip. This PROM, referred to as firmware, is then incorporated into the final product. If the software development for this firmware constitutes a significant portion of the proposal, it should be pointed out to the ACO/PCO as an area worthy of technical analysis.

A basic premise of software estimating is that the effort should be decomposed into functions (modules) to be accomplished before an estimate of the lines of code (LOC) is created. Generally, the finer these functions are broken down, the more accurate the LOC estimate will be. REVIC allows the user to enter as many modules as are decomposed from the proposal.

When entering the modules into REVIC the user must decide if the modules will be developed independently of each other. This matters because the relationship between the lines of code and man months of effort is not a linear one. Therefore a project with several modules developed dependently takes proportionally more time than the same size modules developed independently. For example, the REVIC effort equation for the embedded mode is:

$$MM = 3.312(KDSI)^{1.2}$$

where MM = Man Months of effort and
KDSI = Thousands of Delivered Source Instructions (DSI) (or LOC).

One 14,000 DSI project would require 79 MM, while another project broken into 7 independently developed modules of 2,000 LOC would require 7.6 X 7 or 53 MM.

Highly interrelated modules developed by one group of programmers and analysts should probably be entered in the same REVIC session. Modules with a low degree of interface developed somewhat more independently should be run in discrete REVIC sessions with the resultant man month estimates added manually.

The project must be viewed as an amalgam of the whole when deciding upon the values of various input parameters. Even though there might be some variation of talent among programmers, for example, judgements about the overall level of talent must be made. There is no provision in REVIC, at this time, for assigning different input parameters to different modules unless the modules are being developed independently. In that case, different parametric values may be assigned using discrete REVIC sessions.

C. REVAD Features. REVAD is designed for use with little training. Read each screen carefully and answer the questions. If any question is answered with a response of "unknown" (specified by entering a question mark (?)), REVAD will supply a default value for that parameter.

1. Automatic Link to REVIC. If you use REVAD to determine the values needed for REVIC, you do not need to manually re-input these values to REVIC. The Advisor produces a data file which can be called up by REVIC. The name of this file is supplied by the user with the file extension of .RDF. The chosen name should be unique for each project to avoid any confusion. This file can be edited in REVIC if you want change some parameters for "What if" types of analyses.

2. Automated Report (TACP Addendum). The Advisor creates a report for use as an addendum to the TACP. This report, begins by explaining the use of parametric models for software cost estimating. It then lists all of the REVIC parameter input values and the rationale for their selection. You may edit this report with most word processing packages. This report carries the same user specified name with a file extension of .WPF.

3. Glossary. This is a dictionary of nearly 100 software engineering related terms. Whenever a REVAD menu displays "G - Glossary" as an option, the user may press a "G" to access the dictionary.

II. INSTALLING REVAD. The software provided to run REVAD also includes REVIC. All of these files, as well as the automatic link file, will reside on the hard drive of your PC. Basic requirements are an IBM PC, PC-XT, or 100% compatible PC running DOS 2.0 or above. REVAD is memory intensive. It was designed to run on those machines with at least 640k RAM available. Make sure there are no resident programs, such as SideKick, in memory while using REVAD.

When an instruction requires the user to press the enter key, this guide will use the convention: <enter>. The double quotes " " highlight the required keystrokes. DO NOT type the actual quote marks.

1. In the root directory, create a subdirectory for the REVAD programs.

   a. At the C> prompt, type "md revad" <enter>.

b. The C> prompt will return. Type "cd\revad" <enter>.

2. Copy the necessary file to the newly created REVAD subdirectory.

a. After step 1b, the prompt reads: C:\REVAD>. Insert the REVAD disk and type: "copy a:installr.exe" <enter>.

b. After the file is copied from the floppy disk, make sure you are still at the C:\REVAD prompt, and type "installr" <enter>. Wait for the installation process to finish.

c. To make sure you have all the files you need, type "dir" <enter> at the prompt (which should read: C:\REVAD>). The files in the REVAD subdirectory should be (the order doesn't matter):

VPXRUN.EXE
VPX.MSG
REVADVP0.KMP
REVADVP1.KMP
REVADVP2.KMP
REVADVP3.KMP
REVADVP4.KMP
REVADSCR.TXT
REVGLOSS.TXT
REVADFMT.EXE
RPTWORK.DBF
REVADDBF.DBF
DEFAULT.RVS
DEFAULT.LOC
REVIC.EXE
RV.BAT
INSTALLR.EXE

3. You are now done installing REVAD (as well as REVIC), on your PC.

Note: After installation, you may delete the installation program "INSTALLR.EXE" from the hard drive.

III. STARTING REVAD. When an instruction requires the user to press the enter key, this guide will use the convention: <enter>. The double quotes " " highlight the required keystrokes. DO NOT type the actual quote marks.

1. To use REVAD you must be in the REVAD subdirectory. Unless you are already in the REVAD subdirectory (prompt will be C:\REVAD>), type "cd\revad" <enter>.

2. At the C:\REVAD> prompt, type "rv" <enter>.

3. Now you are ready to use REVAD. Questions are answered by selecting a choice from a menu of responses, or (in some cases) by entering an appropriate word or value in response to an open-ended question. In addition, each screen may indicate special keys which are active and which initiate special functions (for example, entering the glossary, saving the session, etc.). These special keys are further described in Paragraph E below.

IV. USING REVAD.

A. GENERAL. A session of REVAD can be thought of as a series of individual sections. Each section determines a value for one of the parameters for input into REVIC. The value for each parameter is deduced from answers you supply to a series of questions. Each section proceeds along the same general pattern: first, one or more screens provide explanatory information and guidelines for answering the questions; next, you must supply answers to questions which will be used to determine the parametric value; and finally, the value of the parameter is displayed, along with the question "Are you satisfied with the value?" of the parameter. Responding to the question with a Yes completes the section for that parameter, and brings the next section. Responding with a No (or unknown) restarts that section, beginning with the explanatory screens. The sections appear in a pre-defined order. You may review and change any of parameter choices again at the end of the session. A section is not complete until you respond Yes to the question "Are you satisfied with the value?".

When starting REVAD, you have three options. You may (1) begin a completely new session, (2) call up a previously saved session and resume it at the point you left off or (3) choose the default session with the associated default settings of all parameters and skip right to the end of the session. By simply modifying the default parameters, rapid "what if" sessions may be conducted. While "what if" sessions may also be done in REVIC, REVAD automatically produces the automated report, described in section I.C. above.

Note: Whenever you begin a new session of REVAD, or begin a session using the default values, you are first prompted to provide a session name. Any files saved from the session are prefixed with the session name, and REVAD automatically adds an appropriate file extension for each type of file saved. The session name must NOT contain a file extension, and must conform to standard DOS file naming conventions (8 characters or less, no embedded spaces, etc.). For best results, form the session name using only the characters A through Z and the digits 0 through 9. Upper-case and lower-case letters are equivalent. See the section on REVAD FILES below for further information on the different types of files.

Caution: Any files saved by REVAD overwrite existing files with the same name. Use care to ensure your session names (and hence your filenames) are unique.

B. SAVING A SESSION. If you need to interrupt a REVAD session for any reason, you may save the data entered up to that point by pressing an "S" whenever "S - Save" is indicated as an option towards the bottom of the screen. The data is automatically saved under the name you provide as the session name, with a file extension of RVS. At the same time, the lines of code data is also saved with the same session name, but with a file extension of LOC. After the data has been saved, you may continue with the session. Saving a session does not automatically exit REVAD.

C. RESUMING A SESSION. Following the introduction of a REVAD session, you are asked if you wish to begin a new session, resume a previous session, or start with default values. If you choose to resume a previous session, you are asked to provide the name of the old session. REVAD then indicates whether or not the session data was found. If it is found, you are asked to verify whether this is the correct session (by responding either Yes or No). If you respond Yes, the data is loaded, and you are placed at the point of REVAD where you left. If you change your mind and respond No, the series of questions repeats, again asking if you wish to begin a new session or resume a previous session. You may also restart the questioning by entering the word CANCEL when prompted for a filename.

D. EXITING REVAD. You may exit REVAD completely and return to the DOS level prompt by pressing the "X" key whenever the item "X - Exit w/o save" appears towards the bottom of the screen. You may need to move forward a screen or two before this item appears, but this does not interfere with the completed portions of the session. Exiting REVAD in this manner does NOT save any data from the session. If you intend to resume the session at a later time, then press the "S" key (to save the session) before pressing the "X" key (to exit the session).

E. SPECIAL KEYS. REVAD uses a full screen display. On most screens, a menu line at the very bottom of the screen provides information on special keys which are active for that screen. The ones you will encounter are explained below. Not all keys are active for all screens.

S - Skip introduction	Press S to skip the introductory material.
G - Glossary	Press G to display the glossary.
R - Return to session	Press R to quit the glossary and return to the consultation screen.
B - Back up	Press B to display the previous screen.
S - Save	Press S to save the session up to the current point.
X - Exit w/o save	Press the X key to immediately exit REVAD and return to the DOS prompt. Nothing from the session is saved. If you wish to save the session data up to the current point, then press S before pressing X.
Any other key to continue	Press any key other than the special keys indicated to go on to the next screen.
Press any key to continue	Self-explanatory.
Arrow keys	Use the arrow keys to move through a list of response choices where appropriate.
?	Responding to a question with a question mark indicates "unknown". A default value is then provided.
ENTER	Use this key to select a highlighted choice, or to terminate a response to an open-ended question.
BACKSPACE	Use the BACKSPACE kev to erase characters on open-ended questions.
Other keys	Avoid using the special keys, such as the function keys, insert and delete keys, page up and page down, ESC, etc. They produce undesirable results.

<u>CAUTION</u>: Do NOT press the "/" during a REVAD session as you may exit the advisor without saving your work!

F. MISCELLANEOUS.

1. Selecting from a list of responses to a question.

Use the arrow keys to move through the various choices provided as responses to a question, and then press <ENTER> to select a choice. You must use the arrow keys to move through the various choices.

2. Responding to a question with "unknown".

Enter a question mark (?) to indicate an "unknown" response to a question, if appropriate. A response of "unknown" is never explicitly provided as one of the possible response choices, but it is always implicitly available. For the best possible determination of a parametric value, however, each question should be answered as best as possible.

Note: You may not respond to an open-ended question with "unknown", as only specific types of values are accepted. As an example, when responding to a question asking for the size of the database in bytes, a numeric value must be entered. A response of "unknown" will not be accepted.

3. Glossary.

Display the glossary by pressing the "G" key whenever "G - Glossary" is indicated as an optional response. You then type in the first letter of the word you want. The first word beginning with that letter will display on the screen. You may then move through the glossary forwards (by pressing any key) or backwards (by pressing the "B" key) one page at a time. Return to the session by pressing the "R" key.

4. Restarting REVAD.

After a completed session, you may restart REVAD using the current session values by pressing "U" in response to the item "U - Restart REVAD using current session values." You are then placed in the screen where you can selectively change any of the environmental values or the session name. Of course, you can always start REVAD by typing "RV" at the DOS level prompt, and resume an old session by supplying the appropriate session name.

G. <u>REVAD FILES</u>. All files saved by REVAD are pre-fixed with the name you provide as a session name at the beginning of a session. This name remains in effect unless it is changed at the very end of the session. Four types of files may be saved by REVAD. Each is identified by a unique file extension, and each is explained below. For illustration, assume the session name is SAMPLE.

SAMPLE.RVS - Files with a file extension of RVS contain the REVAD session data. An RVS file is automatically saved whenever you press the "S" key in response to the item "S-Save" (see SAVING A SESSION above).

SAMPLE.LOC - Files with a file extension of LOC contain the lines of code data. An LOC file is saved at the same time the RVS file (above) is saved.

SAMPLE.WPF - Files with a file extension of WPF are the TACP addendum word processing files. This file is saved whenever you select SAVE AND QUIT at the very end of a session. A WPF file is not saved unless a session is complete.

SAMPLE.RDF - Files with a file extension of RDF contain the values for input into REVIC. This file is saved at the same time the WPF file (above) is saved.

Note: You may delete old files using the DOS DELETE command. If you delete the RVS file, always delete the corresponding LOC file, and vice versa. This will prevent REVAD from erroneously trying to use one file without the other, as these two files are always used as a pair.

V. REVAD QUICK REFERENCE GUIDE. This section is a quick "How To" reference for typical functions in REVAD, <enter> will be used to mean the "ENTER" key. Double quotes (" ") will be used to highlight the keystrokes required. Do not actually type the quotes when entering the keystrokes.

HOW DO I START REVAD?

From the REVAD subdirectory, type "RV" <enter>.

HOW DO I EXIT REVAD BEFORE FINISHING A SESSION AND SAVE THE DATA TO THAT POINT?

1. Press the "S" key when "S - Save" appears as a menu item near the bottom of the screen.
2. Press the "X" key to exit REVAD and return to DOS.

HOW DO I RESTART REVAD AFTER QUITTING DURING A SESSION AND SAVING THE DATA?

1. From the REVAD subdirectory, type "RV" <enter>.
2. Following the introductory screens, you are asked if you wish to begin a new session, or resume an old session. Use the arrow keys to highlight the choice "resume old session", and press <enter>.

3. At the next prompt, enter the name under which the session was saved (or enter the word CANCEL if you change your mind).

HOW DO I EDIT THE AUTOMATED TACP REPORT ADDENDUM?

1. The report is saved as a flat ASCII file, with a unique file name (specified by you) with an extension of .WPF.
2. Edit the .WPF file with your word processor. Make sure you include the path C:\REVAD\ when calling up the file.

HOW DO I START REVIC AND CALL UP THE REVAD OUTPUT FILE IN REVIC?

1. From the REVAD subdirectory, type "REVIC" <enter>.
2. Using the arrow keys, place the cursor on Begin, then press <enter> twice.
3. You will see the message "Do you want to load a previously saved data file? (Y/[N]). The [] indicate the default response, in this case N.
4. Type "Y" and a list of all files in the REVAD subdirectory with an .rdf extension will appear on the screen. Then simply highlight the correct file (using the arrow keys) and <enter>.

HOW DO I CHANGE REVIC INPUT FILE WITHOUT RUNNING REVAD OVER?

In REVIC:

1. Call up the file in REVIC as explained above.
2. To change any of the 18 environmental factors, highlight "Environment", then <enter>. Follow the menu to edit the proper fields.
3. To change line of code factors, highlight "Sloc", <enter>. Follow the menu to edit the proper fields.

In REVAD:

1. Start REVAD, call up completed session.
2. Use the final screen to enter required changes.
3. Save +.Quit will save modified sessions under your old name.

HOW DO I START REVIC FROM A REVAD SESSION?

1. At the very end of a REVAD session, after the data file and word processing report have been created, you are given the choice to either run REVIC (by pressing the "R" key) or return to the DOS level prompt. Type "R" if desired.
2. You may start REVIC from REVAD only after a completed session. Otherwise, you must start REVIC from the DOS level prompt by typing "REVIC" <enter>, as explained above.

REVIC INFORMATION
SECTION II
TABLE OF CONTENTS

REVIC

SOFTWARE COST ESTIMATING MODEL

USER'S MANUAL

VERSION 9.0
April 1991

Developed By
Raymond L. Kile

TABLE OF CONTENTS

SECTION 1 - DESCRIPTION

REVIC EQUATIONS

The REVIC (Revised Intermediate COCOMO) model predicts the development life cycle costs for software development from requirements analysis through completion of the software acceptance testing and the maintenance life cycle for fifteen years. It is similar to the intermediate form of the CONSTRUCTIVE COST MODEL (COCOMO) described by Dr. Barry W. Boehm in his book Software Engineering Economics. The intermediate COCOMO model provides a set of basic equations which calculate the effort (manpower in man-months and hours) and schedule (elapsed time in calendar months) to perform a typical software development project based on an estimate of the lines of code to be developed and a description of the development environment. The forms of the basic equations are shown in Figure 1.

$$(1) \quad MM = A * (KDSI)^{B} * Fi$$

$$(2) \quad TDEV = C * (MM)^{D}$$

FIGURE 1 - COCOMO EQUATIONS

Equation (1) predicts the manpower (MM) in man-months based on the estimated lines of code to be developed (KDSI is an acronym for Thousand Delivered Source Instructions) and the product of a group of environmental factors Fi. The coefficients (A, B, C, and D) and the factors (Fi) are determined by statistical means from a database of completed projects. The factors attempt to account for the variations in the total development environment (such as programmer's capabilities or experience with the hardware/software) which tend to increase or decrease the total effort and schedule. The results from equation (1) are input to equation (2) to determine the resulting schedule (TDEV is an acronym for Time for Development) in months needed to perform the complete development. The COCOMO model then provides a set of tables which provide a distribution of the effort and schedule to the phases of development (system engineering, preliminary design, critical design, etc.) and activities (system analysis, coding, test planning, etc.) as a percentage of the total.

SOFTWARE DEVELOPMENT MODES The intermediate form of the COCOMO model has three basic modes of software development. These modes describe the overall software development in terms of the size, number of interfaces, and complexity. Each mode provides a different set of coefficients for the basic equations. In addition to the three COCOMO modes, REVIC provides a fourth mode specifically for Ada projects. The four modes are described in TABLE 1.

ORGANIC - Standalone program with few interfaces, a stable development environment, no new algorithms, and few constraints. Usually very small programs.

SEMIDETACHED - A combination of organic and embedded features.

EMBEDDED - Programs with considerable interfaces, new algorithms, and/or extremely tight constraints. Usually very large or complicated programs.

ADA - Programs designed and developed using the Ada design, methodology, language, and APSE.

TABLE 1 - COCOMO/REVIC Software Development Modes

REVIC assumes a default of Embedded mode upon startup. If you need to change the mode, you will have the opportunity in the SELECT ENVIRONMENT display screen described later in this manual.

DEVELOPMENT PHASE DESCRIPTIONS The COCOMO model, as represented by equation (1) and (2), is based on the development phase of typical software development activity. The development phase periods are described in TABLE 2.

There are several things to note about Table 2. First, the development period for the COCOMO model does not include the system engineering phase. The amount of effort expended in a system engineering phase can vary dramatically based on how much work was previously done in a Concept Validation phase or during the proposal period. The REVIC program predicts the effort and schedule in the system engineering phase by taking a percentage of the development phases. REVIC provides a default value for this percentage based on average programs, but allows you to change the percentage if desired.

Phase	Start Milestone	End Milestone
System Engineering	Generally Contract Award	Completion of S/W Specification Review (SSR)
*Preliminary Design	Completion of S/W Specification Review (SSR)	Completion of Preliminary Design Review (PDR)
*Critical Design	Completion of PDR	Completion of Critical Design Review (CDR)
*Code & Unit Testing	Completion of CDR	Completion of CSC Testing by the S/W programmers (CUT)
*Integration & Test	Completion of CUT	Completion of the Prelim Qual Test (PQT) at the CSCI level
Development Test & Evaluation	Completion of the Formal Qual. Test (FQT) at the CSCI	Completion of the System Level Functional and Physical Configuration Audits (FCA, PCA)

* - Items are covered in the Intermediate COCOMO equations

TABLE 2 - Development Phase Descriptions

The second item of note in Table 2 is that the COCOMO development phase stops at completion of the integration and test phase (after successful PQT). This phase is characterized by the integration of software CSCs into CSCIs and testing of the CSCIs against the test criteria developed during the program. It does not include the subsystem level integration (commonly called software builds) of CSCIs together or system level testing to ensure the requirements of the system level specifications are met. The S/W to S/W and S/W to H/W integration and testing is accounted for in the Development Test & Evaluation (DT&E) phase. The REVIC program predicts the effort and schedule for this phase similarly to the system engineering phase, by taking a percentage of the development phases. Again, REVIC provides a default percentage for this phase based on average programs and allows the user to change the percentage if desired.

Some contractors combine the Code and Unit Test phase with the Integration & Test phase and call the result the Construct phase. The total effort predicted by the REVIC model for the two combined phases should be added together into the Construct phase if you are using that convention.

MAINTENANCE PHASE DESCRIPTION

The REVIC model provides an estimate for the maintenance of software over a fifteen year period by implementing the equation described in Dr. Boehm's book and shown in Figure 2.

$$\text{(3)} \quad MMAM = (MMnom) * ACT * MFi$$

FIGURE 2 - COCOMO MAINTENANCE EQUATION

In equation (3), MMnom is the result of equation (1) without multiplying by the environmental factors (Fi). The first new term in the equation, ACT, is the annual change traffic expressed in percent. The second new term in the equation, MFi, is a set of maintenance environment factors. REVIC provides a default percentage for ACT and also allows you to change it if desired.

The program assumes that there will be some transition period after delivery of the software during which residual errors will be found before reaching a steady state condition and provides a declining, positive delta to the ACT during the first three years. In the first year, the ACT is multiplied by 1.5 by REVIC, while in the second year the multiplier is 1.3 and the third year's is 1.15. The fourth and all subsequent years apply the straight value specified for ACT.

DIFFERENCES BETWEEN REVIC AND COCOMO

The primary difference between REVIC and COCOMO is the basic coefficients used in the equations and REVICs addition of a mode for Ada programs. REVIC has been calibrated to a data base of recently completed DOD projects and uses different coefficients. On the average, the values predicted by the basic effort and schedule equations will be higher in REVIC vs COCOMO. The Air Force's Contract Management Division published a study validating the REVIC equations using a data base (the data base collected by the Rome Air Development Center) different from that used for calibration. In addition, the model has been shown to compare to within 2% of expensive commercial models (see CALIBRATION AND ACCURACY below). Other differences arise in the mechanization of the distribution of effort and schedule to the various phases of the development and the automatic calculation of standard deviations for risk assessment. COCOMO provides a table for distributing the effort and schedule over the development phases based on the size of the code being developed. REVIC provides a single weighted 'average' distribution for effort and schedule, along with the ability to let the user vary the percentages in the system engineering and DT&E phases. The REVIC model has also been enhanced by using PERT statistical methods for determining the lines of code to be developed. Low, high, and most probable estimates for each CPC (computer program component) are used to calculate the effective lines of code and the standard deviation. The effective lines of code and standard deviation are then used in the equations rather than the linear sum of the estimates. In this manner, the estimating uncertainties can be quantified and, to some extent, reduced. A sensitivity analysis showing the plus and minus three sigmas for effort and schedule is automatically calculated using the standard deviation.

In the area of public domain software, there have been many faithful attempts to mechanize the COCOMO model in spreadsheets, usually Lotus based. All of these efforts suffer from the serious flaw of requiring the user to be extremely knowledgeable about the model and capable in manipulating the spreadsheet. REVIC stands head and shoulder above those attempts in its ability to isolate the user from any required detailed knowledge of the algorithms. Through the use of non-modal programming with extensive prompting and help screens, REVIC removes the typical intimidation factor that prevents analysts from using models successfully.

The majority of the REVIC program is concerned with a user-friendly method of prompting for the information needed by the program to determine the proper coefficients, factors, and distribution tables. In effect, the program provides a complete interface between the user and the model. REVIC further distances itself from other public domain models in providing the capability to interactively constrain the schedules and staffing levels. Schedules can be constrained either in the aggregate or by phase of the development effort. Staffing can be constrained by phase. Using these features, the analyst can estimate cost over/underruns and schedule slips at any major milestone.

Once the initial input process is complete (i.e. the program has enough information to calculate the effort and schedule), the user is presented with a custom menu which allows fully exercising the analytical features and displays of the program. All inputs are easily recalled and changed to facilitate analyses, and in addition, the user can constrain the solution in a variety of ways as described above.

CALIBRATION AND ACCURACY

The level of accuracy provided by the model is directly proportional to the confidence you have in the line-of-code (LOC) estimates and the description of the development environment. When little is known about the project or environment, the model can be run leaving all the environmental parameters and default values at their nominal settings. The only input required is a LOC estimate. As the details of the project become known, the data file can be reloaded into the program and the nominal values can be changed to reflect the new knowledge. In this way, the accuracy of the model continually improves.

REVIC's coefficients have been calibrated to a data base of recently completed projects (development phases only) by using the techniques described in Dr. Boehm's book, and provides estimates which are within 5% of the projects in the data base. The REVIC model was compared to the commercially available System 3 model (Dr. Randal Jensen's model as distributed by Computer Economics Inc.) during a study performed for the Department of Defense. After establishing identical environments for both models, the a erage efforts predicted by both differed by less than 2%, while the average schedules differed by 6%. The primary reason for the greater schedule variance is probably due to REVIC's (and COCOMO's) level step staffing profile compared to the Jensen model's smooth Rayleigh curve staffing.

OTHER REFERENCE MATERIALS

It is strongly suggested that you read Software Engineering Economics by Dr. Barry Boehm to have a fuller understanding of the implications of parametric modeling. In particular, the book gives detailed descriptions of the software development process and the factors that tend to increase/decrease the total effort and schedule. Context sensitive help is available on-line while running the REVIC model to assist you in establishing the proper environmental description. In all cases, the information given is enough to input data and get results, however Dr. Boehm's book gives a more complete explanation with many examples.

SECTION 2 - REVIC FILES

PROGRAM FILES

REVIC is distributed as a single executable program (REVIC.EXE) for IBM and compatible personal computers using the MS-DOS operating system. During the program's execution, a temporary work file is created to store information about the line of code estimates. This file is always called "WORKLOC.CDE" and may be deleted from your directory if found. It will not affect the REVIC program to leave it on your disk. For speed of execution and easier operation, a separate subdirectory should be created for the REVIC program and the data files that will accumulate. Copy the REVIC.EXE file into the subdirectory before starting the program.

(Note: REVIC is automatically copied to the hard drive when the REVIC Advisor (REVAD) is installed as outlined in Section I). The following DOS statements give an example of creating a new subdirectory called "REVAD" and then copying the program from its distribution disk (assumed to be in disk drive "A") to the new subdirectory.

```
C:>MD REVAD
C:>CD REVAD
C:\REVAD>COPY A:REVIC.EXE
```

DATA FILES - .RDF
The REVIC program provides the ability to save your estimates during the course of running the program for reuse later. The files are identified by the '.RDF' extension and are created whenever you save the results of an estimating session. RDF (REVIC Data File) files are ascii files containing all the program's inputs, results variables, and program flags. These types of files can be read into the REVIC program and will result in the exact state the program was in when the file was saved. This allows you to revisit previous estimates and change the inputs. RDF files could also be read by user written programs for specific purposes. The REVIC Programmer's Notes gives a description of RDF file contents.

DATA FILES - .RFF
A second type of file is identified by the '.RFF' extension. RFF (REVIC Factor File) files are ascii files containing the complete set of environmental factors describing the contractor's development environment and certain project factors such as complexity, security, etc. These files allow you to set up default environments that can be read into the REVIC program and save you having to manually specify the environment. For example, you could have a RFF for one contractor or type of project and another RFF for a different contractor or type of project. An analyst would not be required to 'guess' what environment to specify for a given project. He would merely load the appropriate RFF and proceed with the analysis. The program allows you to save the current environment to a RFF file at any time.

DATA FILES - .RCF
A third type of file is identified by the '.RCF' extension. RCF (REVIC Coefficient File) files are files containing the coefficients for the equations used by the program. These files may be provided to tailor the calibration of the model to specific circumstances. If present, these files may be selected by the user during the execution of the REVIC model. When selected, a message will appear on the displays and reports that non-standard calibration coefficients were used.

SECTION 3 - GETTING STARTED

To start the program, type its name at the DOS prompt (>) followed by a RETURN (cr) as follows: > REVIC (cr)

The program will load and present its opening screen giving the version number of the program and a menu of choices that can be taken. The opening screen suggests that users who are not familiar with the model choose the 'Begin' option from the menu. Experienced users may want to skip the beginning screens to get directly to the ability to load a custom environment or a previously saved data file. They may do so by choosing the appropriate menu item from the opening screen. Choosing the 'Begin' option will result in a fixed series of input screens which will ensure that the user enters all appropriate information. The order of this script is described in the remainder of this section. For information on the other menu choices, see Section 4.

LOAD SAVED FILE

The program presents you with screens giving instructions and requesting various inputs. After starting the program and choosing the 'Begin' option from the opening screen, you will be presented with the only unique screen in the REVIC program (all other screens can be recalled through appropriate menu choices). This screen is showed only when starting the 'Begin' script. You are asked if you want to load a previously saved data file (RDF). A message at the bottom of the screen tells you that hitting the RETURN key without entering anything will always default to the choice enclosed within square brackets. In this screen, the default choice is [N], or NO. Once you have run the program and have saved one or more data files, you may want to start out by loading one of them. If that were the case, you would hit the 'Y' or 'y' key. The file would be opened, read, and the model would jump to the results display. If you answered 'Y' to this prompt and there were no previously saved RDF files, the program would inform you that no files were found and allow you to continue. The first screen, just described, is unique because it determines the mode of operation of the program. If you do not have a RDF file to load, you must enter certain amounts of data to get an estimate. Answering 'NO' to the first screen's prompt means that the program will put up a predetermined minimum number of screens, in order, to ensure that sufficient data has been input to make a valid estimate. Answering YES and subsequently loading the RDF file, takes you directly to the result screens. Once at the result screens, you take control of the program and tell it what you want to do, and the order to do it in, by selecting from a menu of choices.

The remainder of this section will assume that you choose 'NO' to the first screen's prompt.

SELECT ENVIRONMENT

The next screen presents you with 24 environmental factors and default variables. The screen is divided into two sections. The upper part of the screen displays nineteen environmental factors and the software development mode. Below that is a group of four variables with their default settings. Use the cursor control keys (arrows, PGUP, PGDN, HOME, END, TAB, SHIFT TAB) to move between fields. The current field selected will be highlighted. When an environment factor field (upper part of the display) has been selected, you can increment the setting by hitting the SPACE BAR. The setting displayed and its value will increment to the next higher rating until reaching the maximum and will then start over at the lowest rating in a circular buffer. The total product of all nineteen factors (the software development mode is not one of the factors, but is shown with a value of 1.00 in this display to avoid confusion) is continuously updated and displayed immediately below the factors. The total effort and schedule predicted by the model is directly proportional to this number. Larger values result in more effort and schedule, while smaller values result in less effort and schedule. Note that not all the factors have larger values for high settings versus low settings. It will be a temptation to play with the factors to make the predicted effort and schedule come out to the RIGHT ANSWER. Avoid that temptation. You should always set the factors to the actual environment to the best of your knowledge. If you don't like the resulting answer then something in the actual environment or size of the program needs to be changed, but don't fool yourself by specifying an environment that doesn't correspond to reality.

The four variables in the lower part of the screen must be manually input. Each field has a specified length. While entering numbers into a field, if you go over the specified length the field will reset to zero. If you enter a number that fits within the specified length but is outside of a normal range of values for that factor (for example - the number of hours in a man-month can't exceed 248), the program will not replace the default value. This limit checking is not performed until you exit the screen by hitting the ESCAPE key. If you entered an improper number and left the screen, when you reenter this screen at a subsequent time, the value displayed will show that the default number is still in place.

For this screen and most subsequent ones, a notice is shown at the bottom of the display describing the function keys that are active. For this screen you have three choices. Selecting F1 will take you to the on-line help file. You will get a specific help screen based on which field was highlighted when you asked for help. The second choice is Selecting the ESCAPE key. This will cause the program to check all fields, limit check the variables, enter the appropriate data into the programs data base, and proceed to the next screen. The final choice is to Select the F9 key. This will result in quitting the REVIC program. All data input will be lost if you choose this action. All of the environmental factors are described in Dr. Boehm's book and in the help file, therefore, they will not be discussed in this manual. Additionally, the four variables are described in adequate detail in the help screens and will also not be discussed.

NEW CODE DESCRIPTION
Entering the estimates for lines of code to be developed is broken up into two main categories. The first is entirely new code and the next screen describes how the program will ask for the lines of new code to be input. This informational screen is only shown when you have chosen the 'Begin' option from the menu. Arriving at the code enter screens by any other mode assumes that you know what the program will ask for. In essence, power users are not handicapped by having to waste time reading this screen since the same information is available from the help screen available when at the next display. Hit any key after reading the information to proceed.

NEW CODE ENTER
The program stores four pieces of information for each code module in the temporary WORKLOC.CDE random access file; the module's name, the low estimate (LE) for lines of code, the high estimate (HE) for lines of code, and the most probable (MP) estimate for lines of code. The four items are prompted for in four fields. Use the cursor control keys to move between the fields and enter d.ta. The data you input is not entered into the program until you select the F2 key as described on the screen. Help is available by selecting the F1 key. The only number that is required to be entered is the most probable estimate, the other fields may be left blank. If the low and high estimate fields are left blank when the F2 key is selected, they will both be defaulted to the most probable estimate. If only the high estimate field is left blank, it will be defaulted to the most probable estimate. If only the low estimate field is left blank, it will be defaulted to zero. Once you are through entering new code modules, select the ESCAPE key to exit the screen and proceed immediately to the adapted code screens, or select the F9 key to quit the REVIC program. All of the information entered is printed in the appropriate report, therefore it is useful to put in descriptive names for the modules if you will be printing a lot of data.

STATISTICAL TREATMENT OF CODE ESTIMATES
The REVIC program goes beyond the COCOMO model by asking for the low and high estimates for the lines of code to be developed. The program calculates an effective lines of code (EDSI) and the

standard deviation (SD) based on the spread between the high and low estimates. Figure 3 shows the equations used.

$$(4) \qquad EDSI = (LE + 4MP + HE) / 6$$

$$(5) \qquad SD = (HE - LE) / 6$$

FIGURE 3 - STATISTICAL CALCULATIONS

The EDSI is converted to KDSI by dividing by 1000 and is the actual number used in equation (1). The standard deviation is used to present a risk analysis display which will be described later in this manual. When the F2 key is selected, the currently displayed data is entered into the program's data base (actually stored in the WORKLOC.CDE file). The fields are then emptied and the module number is incremented in the display. The program is then ready for another code module to be entered. You are limited in the number of code modules that can be entered only by the amount of available space on your disk.

DECOMPOSING CODE ESTIMATES

Some thought should be given to how the software is decomposed and entered into the program. When more than one module are entered, the EDSI and SD are calculated separately for each module. The EDSIs are added together while the SD of the combined modules is calculated using the square root of the sum of the squares method. This means that the uncertainty in the lines of code as reflected in the standard deviation is smaller when the code is decomposed. In other words, the SD for a single module of 100,000 lines of code with 10% uncertainty will be larger than the sum of ten modules of 10,000 lines of code, each with the same 10% uncertainty. Thus it is more accurate to enter 100,000 lines of code as ten, or more, modules than it would be to enter it as one module. There may be times when you want the SD to be high to see what the bounds of your problem are, but remember that the accuracy of this statistical approach suffers.

PRODUCTIVITY VS SIZE OF CODE

The software development productivity, in terms of lines of code developed per man-month expended, is non-linearly, inversely proportional to the total size of the project. In other words, the larger the size of code to be developed, the lower the actual productivity will be. The practical effect of this phenomenon is that if you can divide a 100,000 lines of code development into two independent teams, each with 50,000 lines of code, it will consume less total effort. Currently, you would have to run the model twice with 50,000 lines of code and add the results together (or run it once with 50,000 LOC and multiply the effort results by 2). A future enhancement to REVIC will allow you to enter the two code modules and specify how they should be treated (as independent efforts or added together); but for this version, they will be added together and the results will reflect the lower productivity due to the larger size.

PRODUCTIVITY VS ENVIRONMENTAL FACTORS

The software development productivity, is also directly related to the resulting modifier when all nineteen environmental factors are multiplied together. A lower MDD or modifier rating will result in a higher productivity rating. Figure 4 shows the relationship between productivity, MDD and size of code for the semidetached mode. Organic and Embedded modes would have similar curves with different values.

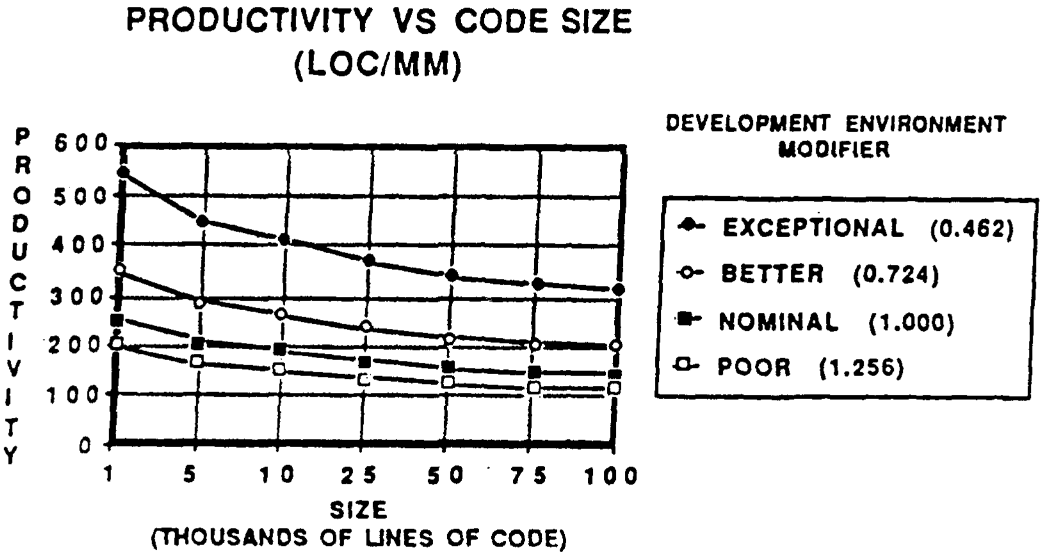

FIGURE 4 - PRODUCTIVITY RELATIONSHIPS

ADA AND PRODUCTIVITY

The exponent used for the Ada mode in equation 1 is less that 1.0 (currently .941 is used in version 8.7.4 and greater). This has unique implications for developments using the Ada programming support environment (APSE) and language.

In general, everything said above concerning productivity vs size is reversed for the Ada mode. Since the exponent is less than unity, the resulting effort tends to be correspondingly less as the size increases and thus the productivity increases with size. The exponent used in REVIC is close enough to unity that the productivity can be considered essentially constant over the range of code sizes and other factors can be used when deciding how to decompose the software. Productivity will still vary with the environment in exactly the same manner as described for non-Ada modes. The phenomena described by the Ada exponent seems to justify the rationale that enforcing interfaces by using the separably compilable specifications vs body allows Ada developments to overcome some of the manpower normally wasted in communicating between design/programmer teams.

ADAPTED CODE DESCRIPTION

The next screen reached after selecting the 'ESCape' key in the new code enter screen describes how the program expects data to be input when you are adapting or modifying an existing software module as part of the overall software development. You are asked if you have any adapted code modules to enter. The default choice available by hitting the RETURN key without entering anything is 'NO'. If you take the default action, the program will go on to calculate with all the data currently entered and display the first results screen. Entering 'Y' or 'y' will cause the program to show the adapted code enter screen. (Actually, entering any character except a "Y" or 'y' will default to the 'NO' option.) Again, this screen is only shown when arriving here via the 'Begin' option from the menu. Power users do not have to suffer this screen every time.

ADAPTED CODE ENTER

The adapted code enter screen is mechanically similar to the new code enter screen already described. Only the types of data being requested are different. The help screens provide a complete description of the requested data, therefore, the information will not be repeated here. Dr. Boehm's book provides a particularly good discussion of adapted code and is recommended reading. The program calculates the Equivalent Delivered Source Instructions (EDSI) in accordance with the equation shown in FIGURE 4. The Adapted Delivered Source Instructions (ADSI) is multiplied by the percents of design modification (DM), code modification (CM), and retesting (IM) required. As an example, an adapted code package which had exactly 100% design modification, 100% code modification, and 100% retest required would result in the EDSI being equal to the ADSI. For statistical purposes, REVIC assumes a 10% (a total of 20% of the calculated EDSI) standard deviation for the adapted code modules. The means of entering the data into the program's data base and exiting the screen are identical to those described for the new code enter screen.

$$EDSI = ADSI * (.4\ DM + .3\ CM + .3\ IM) \ / \ 100 \tag{6}$$

FIGURE 5 - Adapted Code Calculation

PHASE DISTRIBUTION DISPLAY

The next screen display marks the end of the predetermined path started when you chose the 'Begin' option when starting the REVIC program. The phase distribution display shows the predicted effort, schedule, and required manpower distributed by the development phase (system engineering, preliminary design, etc.). From this point on, you will choose your actions from a menu of choices at the bottom of the screen. Besides changing any of the previously input data, you have the opportunity to perform trade-off analyses by constraining manpower or schedules, or both. The power of REVIC goes beyond merely estimating a program's cost. REVIC can, and should, be used as an on-going tool to evaluate the program's progress. Using the ability to specify schedule or manpower by specific phase of the program allows you to determine such things as the effects of understaffing on cost/schedule, the costs of schedule compression, or the expected overruns due to line of code growth. Changing the environmental factors allows you to quantify the results of staffing with new hires when the program was bid based on experience people or vice-versa. The list of ways in which the REVIC program can be used to analyze programs is limited only by your imagination.

SECTION 4 - PROGRAM OPTIONS

USING THE MENU

Once you have reached the phase distribution display or you choose to use the other choices from the menu at the startup screen, further use of the model is directed by your use of the menu displayed at the bottom of the screen. The top line of the two lines displayed is the main menu. Under it are the sub items which are available from the main menu item which is currently highlighted. You may use the cursor arrows to move between the main menu items and see the full set of subitems available. When you have highlighted the main menu item you wish to select, select the RETURN key. You may also select the main menu item directly by hitting the first letter of the main menu item. After you have selected a main menu item, the main menu line is hidden and the submenu becomes active. Choose an item from the submenu in the same manner, with the cursor keys and RETURN key or by selecting the first letter of your menu choice. If you make a mistake or change your mind at the submenu level when using the cursor keys, you may return to the main menu by selecting the ESCAPE key. Each menu and suboptions are described in the following sections.

THE FILES MENU
The first item in the main menu is the FILES menu. Under the FILES menu are choices which allow you to save REVIC Data Files (.RDF), load previously saved RDF files, and restart the program.

FILES MENU - Save Data File
When you choose the Save Data File option, the program will prompt you to enter a filename in which to store the current data. The filename should follow the normal DOS naming conventions. Any letters after eight characters will be ignored. The program will open a data file in the current subdirectory, write the information to it, close the file, and return to the main menu display. If you select the 'RETURN' or 'ENTER' key without entering anything, the program will cancel this action and return to the main menu display.

FILES MENU - Load Data File
When you choose the Load Data File option, the program will put up a list of all the RDF files that exist within the default or last specified subdirectory. You can then use any of the normal cursor control keys to select a file from the list. Selecting the 'RETURN' or 'ENTER' key will cause the program to open and read the filename that is highlighted as a result of your cursor selection. Selecting the 'ESCape' key will cancel this action and return to the main menu display. If there are no RDF files in the current subdirectory, a message will be displayed and the program will return to the main menu.

FILES MENU - Restart
The Restart option will reinitialize all variables and return to the startup screen.

THE REPORTS MENU
The second item in the main menu is the REPORTS menu. You have the choice of getting a full report, an abbreviated phase display only report, a maintenance report, and a CDRL page estimate report. Besides using the REPORTS menu, you can always use the SHIFT- PRT SCRN method of getting a print out of whatever is currently displayed on the screen. Currently REVIC only supports line printers.

REPORTS MENU - Full Report

The Full Report choice will give you a print out containing the phase distribution display, the activity distribution display, the risk display, the environmental factors with their current settings, and the line of code estimates used in the calculations. You will be prompted for a name to identify the report and be asked to put the printer on-line before proceeding. The printed report will have the current date and time from the system clock for reference. If you have entered a session name in a previous report and do not need to change it, you can hit the RETURN key without entering anything. The previously entered session name will be used. If you choose to enter a new name, it will become the default name for all subsequent displays and reports until you choose to change it again. The REVIC program will respond differently to printer problems depending on what type of computer and version of DOS you are using. It may crash the program if you are not on-line when the program attempts to start printing, so heed the warning message and set up your printer before allowing the program to continue (it is a good idea to save the data to a RDF file first) or cancel the print job. After printing the report, the Phase Distribution Display will be shown on the screen along with the main menu.

REPORTS MENU - Phase Report

This option prints out only the Phase Distribution Display and the environmental factors. You are prompted for a session name and to make sure the printer is on-line in a similar manner to the Full Report option. After printing the report, the Phase Distribution Display will be shown on the screen.

REPORTS MENU - Maintenance Report

This option prints out the fifteen year maintenance projections. You will be prompted for the session name and asked to put the printer on-line. The report will have the current date and time for reference. Maintenance reports show the total effort in manmonths, the average number of full time support personnel required, the average inflated labor rate, and the extended total costs per year for the direct labor.

REPORTS MENU - CDRL Page Estimates

This option prints out the page estimates for DOD-STD-2167A type documents. The prompts for session name and putting the printer on-line are identical to those described for the other reports. You will be given the approximate page estimates for most of the typical types of documents. The listing does not mean to imply that one of each type document must be produced and is costed in the predicted effort. The program is calibrated to the completed programs in a data base, and therefore represents an average amount of documentation. If your program has an abnormal amount of documentation required, you might want to increase the effort over that predicted by the model to accommodate your special circumstances.

THE ENVIRONMENT MENU

The third item in the main menu is the ENVIRONMENT menu. You have the choices of editing the environmental factors, loading a previously saved RFF file, saving the current factors to a RFF file, loading a previously saved RCF file, or specifying the Ada factors.

ENVIRONMENT MENU - Edit Factors

This option will put up the Select Environment screen described in SECTION 3, GETTING STARTED. You may change any of the settings or variables within this screen. After exiting the Select Environment screen, the program will calculate the new results and put up the Phase Distribution Display.

ENVIRONMENT MENU - Load Factor File

This option will cause the program to search the current default subdirectory for any REVIC Factor Files (RFF). A list of RFF files will be displayed and you will be able to use the cursor control keys to highlight the file you want to load. When you have highlighted the right file, select the ENTER or RETURN key to select it. The program will open the file and read in the data. After closing the file, the program will put up the Select Environment screen to show what was loaded. After exiting the Select Environment screen, the program will recalculate the results and put up the Phase Distribution Display. Any forcing, such as forcing the manpower by phase or the total schedule, will be lost and only the results of the nominal calculations will be displayed. Hitting the ESCape key will cancel the action and return you to the main menu. If there were no files in the current subdirectory, a message will be shown and the model will put up the Select Environment screen with the current settings.

ENVIRONMENT MENU - Save Factors

This option allows you to save the environmental factors, including the program default variables and inflation rates, to a REVIC Factor File. The program will prompt you to input the filename to use, then will open the filename under the current default directory, write the data to the file, close the file, and then return you to the main menu. If you enter an improper filename, the program will display a message and give you an opportunity to correct it. Selecting ENTER or RETURN without entering anything will cancel this action and return you to the main menu.

ENVIRONMENT MENU - Get Coefficient File

This option will cause the program to search the current default subdirectory for any REVIC Coefficient Files (RCF). A list of RCF files will be displayed and you will be able to use the cursor control keys to highlight the file you want to load. When you have highlighted the right file, select the ENTER or RETURN key to select it. The program will open the file and read in the data. After closing the file, the program will recalculate the results and put up the Phase Distribution Display. Any forcing, such as forcing the manpower by phase or the total schedule, will be lost and only the results of the nominal calculations will be displayed. Selecting the ESCape key will cancel the action and return you to the main menu. If there were no files in the current subdirectory, a message will be shown and the model will return to the main menu.

ENVIRONMENT MENU - Ada Use
This option puts up an informational screen which tells the user that Ada has been implemented as another mode in the Edit Factors screen.

THE CONSTRAINTS MENU
The fourth item in the main menu allows you to force the model to accept a schedule or manpower loading of your choice. When you force the model, it recalculates the effort and schedule according to whatever constraining condition you have input. Note that shortening the schedule will result in a lower setting (with a higher numeric value) for the schedule environmental factor, with a proportional increase in the total effort to develop the software.

CONSTRAINTS MENU - Remove Constraints
This option will remove all schedule or manpower constraints applied through the CONSTRAINTS menu. It will recalculate and display the nominal results in the Phase Distribution Display. After choosing this option, or before applying any constraints, the Phase Distribution Display will reflect the schedule is NOMINAL. Any other value for the schedule in the display indicates that the program is still in constraint mode.

CONSTRAINTS MENU - Total Schedule
This option allows you to force the model to accept a total schedule of your choice. You will be given a message telling you that the model will not allow you to specify a schedule that is shorter than approximately 75% of the nominal schedule it calculated. The program will tell you what that value is and will then prompt you to input your desired total schedule. If you select the ENTER or RETURN key without entering anything, this action will be canceled and you will be returned to the main menu. After inputting your desired schedule, the program will iterate its calculations to arrive at a solution that meets your desired schedule within the iteration criteria of one tenth (.1) of a month and display the results in the Phase Distribution Display. The display will automatically change the value of the schedule actor and display the current setting.

If you enter a total schedule that is too short, the program will put up an error message and tell you to reenter. The value shown for the minimum schedule is rounded off for display, therefore you may need to enter a number that is slightly higher if you are trying to get the shortest possible schedule.

The minimum possible schedule is maintained properly even when you choose to change the system engineering or DT&E schedule through the MISC menu. The minimum possible schedule is based on the development phase portion of the entire lifecycle. You can shorten the total schedule by zeroing out either of the two non-developmental phases, but you will still not be allowed to shorten the nominal development phase below the minimum.

CONSTRAINTS MENU - Manpower By Phase

This option allows you to specify the number of full time personnel in each phase of the total lifecycle. The model will put up a display of what is currently in each phase. Use the cursor control keys to move around and change the values as desired. When you tell the program to enter the data by selecting the 'F2' key, the schedule will be calculated. If the resulting schedule is below the 75% minimum schedule, you will be given an error message and allowed to change your inputs. Selecting the ESCAPE key will cancel this action and tell the program to recalculate the nominal values. Note that this option allows you to perform a type of cost to complete analysis. If you are at one of the major milestones, you can force the model by putting in the actual manpower used up to that point. The Phase Distribution Display will show what the effect on the total effort and schedule will be (assuming that the program is staffed according to the nominal levels for the rest of the program). You can also put in any proposed under or over staffing and determine the total effect in a similar manner.

CONSTRAINTS MENU -Schedule By Phase

This option is similar to the Manpower By Phase item except that you are specifying the actual schedule. The program will check against the 75% minimum schedule and then recalculate the effort and personnel needed. This option allows you to perform a similar analysis to the Manpower By Phase option. In this case, you can force the model by inputting the actual schedule through the major milestones and let the model tell you the total effect on effort and total schedule.

THE MISC MENU

The next item in the main menu allows you to change the defaults used in calculating the system engineering and DT&E phases, allows you to specify the inflation to be used in calculating the fifteen year maintenance effort, and will allow you to change the phase distribution percentages used in allocating portions of the effort and schedule to the phases of development (system engineering, preliminary design, etc.).

MISC MENU - System Engineering
This option allows you to change the default percentages used in calculating the effort and schedule for the system engineering phase of the lifecycle. Use the cursor control keys to move between the fields and enter your new values. Selecting the 'F2' key enters the data and tells the program to calculate the results and put up the Phase Distribution Display. Entering a zero in the percentage field will zero out the system engineering phase. Selecting the ESCAPE key will return the system engineering values to the original defaults. The value to be input for the schedule should be the actual months to be used by the model.

MISC MENU - DT&E
This option is similar to the System Engineering option. In this case you are changing the values for the Development, Test, and Evaluation phase of the lifecycle. Note that the value for the DT&E percentage set in the Select Environment display changes the default value that is used in the calculations.

MISC MENU - Inflation Factors
This option allows you to specify the inflation factors to be used in the fifteen year maintenance calculations. Use the cursor control keys to move around and change the values. Selecting the 'F3' key will reset all fifteen values to zero. Selecting the 'F4' key will replace all fifteen values with the value that is currently in year 1. Selecting the ESCAPE key will enter the values and return you to the main menu. Note that the inflation factors are saved in the REVIC Factor File. Upon program startup and after a restart, the inflation factors are all set to zero.

MISC MENU - Phase Distributions
This option is not implemented in this version of REVIC. After hitting any key, you will be returned to the main menu.

THE DISPLAY MENU
The next item in the main menu allows you to choose between various displays showing the results of all calculations.

DISPLAY MENU - Phase Dist. Display
This option will recalculate all results and put up the Phase Distribution Display. This display shows the effort, schedule, staffing, and costs by phase of the lifecycle.

DISPLAY MENU - Risk Display

This option will put up the Risk Display. This display uses the standard deviation calculated from the lines of code estimates to show the plus and minus three sigma values for effort, schedule, and cost. Use this display to determine the risk position based on the proposed cost for the software development.

In a 'fixed price' environment, you would expect the proposed cost to be on the high side of the nominal values, since the contractor is assuming all of the risk. You would normally not expect the proposed costs to be above the three sigma values. In a 'cost plus' environment, you would expect the proposed cost to be on the low side of the nominal values, since the government is assuming most of the risk. If your estimates and the proposed costs vary greatly, you should investigate the possibility that your basic assumptions concerning line of code estimates (and the standard deviation) or the proposed development environment are incorrect.

DISPLAY MENU - Activity Display

This option puts up a display showing the effort in the system engineering and development phases distributed by the types of personnel doing the work. Distributions of effort in the DT&E phase are not shown. See Dr. Boehm's book for a discussion of the types of activities and limitations of this display.

DISPLAY MENU - Maintenance Display

This option puts up a display showing the effort, number of personnel required, and costs for maintaining the software over a fifteen year period. The costs shown are based on the wrap rate entered in the Select Environment screen and inflated by the yearly inflation rate. Selecting the 'F2' key will print a report of the maintenance display similar to that obtained by using the 'Maintenance Report' option of the REPORTS menu. To change the inflation factors, use the 'Inflation Factors' option of the MISC menu. To change the labor rates, use the 'Edit Factors' option of the ENVIRONMENT menu.

DISPLAY MENU - CDRL page estimates

This option will provide a display showing an approximate page count for the various types of CDRLs specified in DOD-STD-2167A. See the discussion of the CDRL estimates under the REPORTS MENU.

THE SLOC MENU

This menu allows you to enter new estimates for lines of code or to edit the existing estimates already entered.

SLOC MENU - Both New & Adapted Code

This option allows you to completely respecify the code modules used in the program's calculations. It completely erases the previous data while all other program variables such as the environmental factors are retained. The program will first put up the new code enter screen. When you select 'ESCape' the adapted code enter screen is put up. The introductory information screens described in Section 3 are not shown. All processing within the screens is exactly as already described in Section 3.

SLOC MENU - New Code Only

This option is similar to the Both New & Adapted Code option except that it will only put up the new code enter screen.

SLOC MENU - Adapted Code Only

This option is similar to the New Code Only option except that it will only put up the adapted code enter screen.

SLOC MENU - Edit Code Estimates

This option allows you to change or delete existing line of code estimates. You can also add additional new code or adapted code modules using this option. You are asked if you want a print out of the existing line of code modules. If you chose the 'Y' option, the program will ask you to put the printer on-line and will then print out all the new code and adapted code modules that are active.

All activity in the edit module is controlled by function keys. You have the option of editing an existing code module (F3), adding a new code module (F7), adding an adapted code module (F8), or leaving the edit module (ESCAPE). If you choose to edit an existing module, the program will ask you to specify the index number. Once the data is displayed on the screen, you can edit it as previously described for the new or adapted code enter modules. You also have another function key active. The F5 key will delete the displayed module from the data base. If you select the ESCAPE key while editing a module, the program will not make any changes to the module in the data base before leaving the edit module. Entering an index number that is out of the range of the currently entered number of modules or was deleted will result in an appropriate message.

If you choose either the add new code or add adapted code function keys, the program will display the appropriate input prompts and allow you to enter the information as previously described in the enter new module routines. If you select the ESCAPE key while entering a new module, the data will not be entered into the data base.

THE BEGIN MENU
This menu item provides a scripted method for new users to enter all data necessary to get an estimate. See Section 3 for a thorough discussion. The submenu item associated with this choice provides a descriptive message to direct a new user how to proceed.

THE QUIT MENU
This menu allows you to quit the REVIC program. The submenu item associated with this choice provides a descriptive message telling you to hit the Return key to quit or to hit the ESCape key to return to the main menu.

ADDITIONAL MENU CHOICES
The REVIC program provides two additional menu choices which are not shown in the mainmenu or associated submenus. These choices are reached by hitting an ALT key sequence anytime that the main menu is active as described in the following sections.

CHANGE PROGRAM COEFFICIENTS
The coefficients and exponents used by the REVIC program in performing its basic calculations may by changed in this screen. To enter this screen hit the ALT "@" key (actually, just use the unshifted "2" key). You will be presented with a display showing the eight equations, effort and schedule for the four modes, and a descriptive title. Use the cursor keys to move around and change any coefficients or exponents desired. Hitting the "F2" key will enter the data and change the equations in program memory. The new equations will remain in memory only as long as the program. If you quit and start REVIC again, the equations will revert to their nominal values. The descriptive name will be displayed on all results to indicate that nonstandard coefficients were used, therefore you must type in a new descriptive name whenever changing the coefficients. The program will substitute the name "Non-Standard Coefficients" for you if you do not change it yourself. A function key is provided to allow reverting to the standard coefficients at any time. Hitting the "F3" key will revert to the standard equations and return to the mainmenu. Hitting the ESCape key will return to the main menu without changing the coefficients currently in memory.

SAVE PROGRAM COEFFICIENTS
The coefficients currently in memory, along with the descriptive name can be saved to a .RDF file by hitting the ALT "*" key (actually the unshifted "8" key). You will be presented with the

screen asking you to name the new file. The coefficients and descriptive name will be saved to the file and you will be returned to the main menu. The stored file can be opened at any time through the ENVIRONMENT menu.

SECTION 5 - REVIC EVOLUTION

CALIBRATION

The REVIC model is the subject of an on-going effort to calibrate the total effort and schedule estimates within the DOD community. All users of the model are asked to participate in the data collection process. A sample data collection sheet is provided as an attachment to this user's manual. The sheet asks for the minimum set of data necessary to calibrate the coefficients and factors of the model's equations, and should be filled out after the DT&E phase of the software development is completed. Completed data sheets should be sent to the office that provided the REVIC model.

SUGGESTIONS FOR IMPROVEMENTS

Please send any comments or suggestions for improvement to:

Gordon Wright
Revic User's Group
Naval Ocean Systems Center
Code 9202
271 Catalina Blvd
San Diego, CA 92152-5000
FAX: (619) 553-6249

REVIC BACKGROUND INFORMATION

THE HISTORY OF REVIC AND

THE REVIC USERS' GROUP

THE REVISED INTERMEDIATE COCOMO (REVIC)

The Revised Intermediate COCOMO (REVIC) software cost estimating model, in its current state and status, has a two-pronged history - it is the result of a meeting of two sequences of events.

First, Ray Kile, the developer of the REVIC model, was working at Hughes Aircraft, Aurora, CO, as a software development manager. He began working with parametric models for software costing around 1983 as a means of getting better information for managing formal DoD software developments. He acquainted himself with a variety of software cost estimating models, such as Putnam's Software Life Cycle Model (SLIM), Jensen's System 3 and System Evaluation and Estimation of Resources (SEER) models, and RCA's Price-S model. None of these models fulfilled his need for an interactive management tool. Because of this, Ray Kile became interested in developing a model with greater user simplicity. The opportunity to create such a model was offered through his AF Reserve involvement.

The other historical sequence of events began in 1980. The Air Force Contract Management Division (AFCMD), Kirtland AFB, NM, lead several initiatives to improve in-plant oversight of government software contracts and better support of software contract proposal evaluations. AFCMD submitted a proposed study to the USAF Business Research Management Center, Wright-Patterson AFB, OH to develop a software cost estimating methodology that could be used by AFCMD activities.

In 1985, the Business Research Management Center let a contract to Technion International, Inc., Wilmington, DE, to perform this study. The principal investigator for Technion was Dr. Richard Werling. During interviews with key individuals, Technion researchers crossed paths with Ray Kile (Major, USAF Reserve) who was conducting his AF Reserve duty at the Air Force Plant Representative Office (AFPRO) Martin, Denver, CO. AFCMD had tasked Major Kile to help develop a software cost estimating model based on Dr. Barry Boehm's Constructive Cost Model (COCOMO) -modeling methodology. The September 1986 Technion final report focused the needed work for software cost estimating and led AFCMD to sponsor Major Kile's further development of the REVIC model.

By late 1986, the REVIC model had completed its initial development and entered a phase of calibration. At this point, some of the history of cost models at Hughes, El Segundo, CA, becomes significant (particularly the Electro-Optical Application Software Department (EOASD), El Segundo, CA. EOASD has been building software since around 1955, concentrating on embedded software. From the hundreds of completed projects, eight were used as the primary data source for model calibration.

Between 1983 and 1985 much work was done by EOASD in investigating and calibrating the Jensen model (known as SEER within Hughes). The findings were:

- The Jensen model, like Putnam's SLIM and Boehm's COCOMO, appeared to have correctly identified a universal "software law" for effort and schedules, including the "Brooks Law Effect" for short schedules.

- After-the-fact SEER runs on the eight reference projects gave both cost and schedule values within five to ten percent of the actuals recorded for the references.

- SEER was clearly a better tool for EOASD cost estimating than the "bottom-up" or cost estimating relationship (CER) method that had been used in the past.

Many software estimates were made using SEER during 1984-1987 for not only EOASD, but for other portions of the Data Systems Division.

Between 1986 and 1988 REVIC was investigated and calibrated by EOASD. The results of this work were:

- REVIC cost and schedule outputs were within ten percent of both the SEER (Jensen) model results and the actual data for the eight reference projects, once the COCOMO coefficients were adjusted.

- REVIC was easy to learn and provided easily understood outputs rapidly.

- The cost reduction and convenience of a PC-based software cost model was evident. SEER was VAX-based, and required a specialist to operate it.

- REVIC was recommended as a replacement for SEER in the Data Systems Division.

The results of this effort, led by Don Merrill of EOASD, produced a REVIC model calibrated on newer data than the COCOMO data base. The REVIC data was also based exclusively on DoD software developments, whereas COCOMO was not. This calibration (for non-Ada projects) was later validated against data bases from Air Force organizations, such as Rome Air Development Center (RADC), Rome AFB, NY, and Electronic Systems Division (ESD), Hanscom AFB, MA.

In November 1986 Major Kile gave a detailed presentation of REVIC at the Air Force Plant Representative Office AFPRO Mission Critical Computer Resource Workshop, at Lowry AFB, Colorado. Several weeks later, in January 1987, a workshop was held at Det 10, to distribute Dr. Werling's report and make plans to implement the suggestions. Based on availability, cost, and known accuracy, REVIC was recommended for use by AFCMD on a trial test. AFCMD had chosen five AFPROs for the Software Estimating Database (SED) trial.

In August 1987 a conference on the SED project was held in Los Angeles at the TRW AFPRO. Both industry and government trial users reported general acceptance of REVIC as a very useful tool.

In March 1988 Major Kile published a users' manual for REVIC. The guide provides the user, even those without any computer skills, to make full use the REVIC model.

In July 1988, the AFPRO Martin Marietta, Denver, CO, lead engineer, Lowell Simons, closed out the Software Cost Estimating Database Project. He reported that their goal of providing a "Program Management" tool had been completed. He further stated that REVIC was a very useful tool for software budgeting, risk analysis, negotiation and management evaluations.

In late 1988 and early 1989 another REVIC calibration was done by Bob Ness and Major George Hozier (USAF Reserve) of HQ AFCMD, using data from a recently completed Ada command and control project (AFATDS). This calibration produced another mode for the model - Ada mode - in addition to the organic, semi-detached and embedded modes of COCOMO. As data from additional Ada projects become available, it is anticipated that the Ada calibration of REVIC will be refined.

In summary, the REVIC model is now calibrated, tested, and approved for use. In addition to the Air Force, REVIC is being used by many government contractors and other companies, as well as organizations in the Army, Navy, and Marines.

REVIC USERS' GROUP BIBLIOGRAPHY

Commercially Available References:

1. Barry W. Boehm, Software Engineering Economics, Prentice-Hall Inc., 1981.

2. Tom DeMarco, Controlling Software Projects, McGraw-Hill, 1986.

3. Robert B. Grady and Deborah L. Caswell, Software Metrics, Prentice- Hall, Inc., 1987.

4. Capers Jones, Programming Productivity, McGraw-Hill, 1986.

5. William H. Roetzheim, Structured Computer Project Management, Prentice Hall, 1988.

ADDITIONAL REFERENCE MATERIAL
SECTION III
TABLE OF CONTENTS

DETERMINING SOURCE LINES OF CODE

Probably the most difficult factor to enter into the REVIC model (or any software model) is an accurate estimate of the lines of code. If a contractor provides an estimate of the lines of code, you must determine if that estimate is accurate. If the contractor does not provide an estimate of the lines of code, you must determine what the contractor used to create their proposal. With this information, in most cases, you will be able to develop your own estimate.

One basic premise of software estimating is that the effort should be decomposed into functions to be accomplished before you can create an accurate estimate of the lines of code. As a general rule, the finer these functions are broken down, the more accurate the estimate will be.

The REVIC model allows the user to enter as many modules (functions) as are decomposed. These functions are often derived from the Computer Software Configuration Items (CSCI) spelled out in the proposal. It is important to determine if these modules will be developed relatively independently of each other due to the fact that the total lines of codes developed is not linearly related to the man month estimate of the software development effort. If twenty modules, each consisting of 5000 lines of code, are highly interrelated and are being developed by the same team, all twenty modules should be entered in one session of REVIC. If twenty 5K modules having little interface were being developed independently, each CSCI should be entered in a discrete REVIC session and the resultant man months added manually. The two methods produce significantly different man month estimates.

The factors that you will enter can have only one value for the entire program. While some modules may be assigned to parts of the team that have more/less capability/experience, you may not isolate this. You must determine composite factors for the entire program development.

Delivered Source Instructions:

1. Excludes comments, unmodified utility S/W
2. Includes JCL (Job Control Language), format statements, data declarations
3. Includes any test drivers that are developed with same care as final product

REVIC uses the lines of code as the basic sizing parameter. It follows a standard PERT technique and calculates a standard deviation based on the uncertainty in the Lines of Code estimates reflected in the spread between the low and high estimates.

The actual lines of code used in the calculation is statistically derived from the three inputs as follows:

DSI = (Low Estimate + 4*Most Probable + High Estimate) / 6

When entering the Lines of Code estimates, the low estimate may be left at 0. This will result in a larger standard deviation. If you desire to enter only the most probable estimate, the low and high estimates will be defaulted to equal the most probable, resulting in a 0 standard deviation. If the high estimate is left at 0, it will be defaulted if the most probable.

WHEN LINES OF CODE ARE ESTIMATED

At this point you will need to determine whether the estimates are reasonably accurate. Normally, as a minimum, the contractor should have: decomposed the requirements to a fairly low level; used the PERT sizing methodology; and made some analogy comparisons. It is also a good idea to check with the PCO\buying activity to determine if their personnel have made an independent estimate for comparison purposes.

DELPHI METHOD

At least three software experts should estimate the lines of code that will be required for each of the functions in the software package. Each of these experts should make an independent evaluation of the LOC required (possibly employing the PERT technique described above) and then compare the estimates through several iterations to reach a consensus. It is ideal if they can also establish a consensus for high and low estimates.

Even when the contractor has provided DSI estimates it is wise to check on the validity of the estimates. This is done by determining: the amount of memory available to the development team, the expansion ratio of the language, and any contractual constraints on memory reserve. Then using the methodologies described below you may determine if the lines of code that are being proposed will produce object code that will fit into the available memory.

IF LINES OF CODE ARE NOT PROPOSED

If the contractor has not developed a LOC estimate, then he most likely has sized by analogy with a previous software development effort, estimated the sizing based upon available memory available to be used, or estimated the size in object words based strictly on the somewhat subjective concept of number of functions to be performed. If the contractor cannot furnish this type of data, then the whole proposal is suspect.

SIZING BY ANALOGY

If the contractor has determined that a previous program is of similar type, then the contractor may attempt to use that completed program as a basis for sizing the current effort. The following formula is based on a somewhat subjective determination of what constitutes a major function within the software. What constitutes a major function is not important if the definition is applied equally to both the completed and proposed software. Use the following formula to determine the object size (how big the program is once it is compiled and loaded in memory) of the proposed program.

$$I = [A/(F^{0.678})] * N^{0.678}$$

where

A is the size of the analogous software in object words

I is the size of the proposed software in object words

F is the number of major functions performed by the analogous S/W

N is the number of major functions to be performed by the new S/W

SIZING BY FUNCTIONS

Sometimes the contractor will only define the major functions to be performed to derive an estimate of the number of object words. The concept of what is a function is not clearly defined. It could be anything from a simple print routine to the concept of an operating system. It is best to think of major functions as to what the program will accomplish, breaking it down to items such as data collection, algorithm processing, output to devices, etc. When the number of functions have been derived, the following formula may be used to approximate the memory required for the program.

$$M = 0.177 * k * ((N^{0.337} * W^{0.147} /t^{0.770})$$

where

M = Memory size in thousands of words of object code
N = Number of major functions to be performed by the S/W
W = Word size in bits
t = Cycle time of processor in microseconds
k = A constant dependent on application

k values

signal processing - 2.573
interfacing - 2.781
navigation - 3.565
weapon fire control - 4.451
missile fire control - 2.727
communication - 3.412
command and control - 4.046

DETERMINING DSI FROM NUMBER OF OBJECT WORDS

Once the object size has been determined, you may convert this to equivalent lines of source code by using the algorithms developed empirically. What we use for the various languages are as follows:

ADA	5:1	JOVIAL	4:1
FORTRAN	6:1	C	3:1
COBOL	3:1	CMS-2	3:1
PL/1	10:1	ASSEMBLY	1:1
PASCAL	4:1	APL	14:1

The expansion ratio is somewhat dependent on the type of application. However, the research into this is somewhat sketchy so we will stick to these values.

To convert to LOC we will use the following formula:

$$DSI = I/ (1 + P * (E - 1))$$

where

DSI = Deliverable Source Instructions
P = Fractional amount of higher order language (HOL) e.g. .90 HOL and .10 assembly
E = Expansion ratio for the HOL
I = Object Memory size (words)

Reference:

United States Army Management College. Technical Analysis of Contractor Cost Proposals for Embedded Computer Systems - Course Book. Rock Island: Fourth Edition (Revised), January 1989.

REVIC PARAMETERS WITH ASSOCIATED VALUES

Parameter	VL	LO	NM	HI	VH	XH	XX
Programmer Capability	1.42	1.17	1.00	.86	.70	--	--
Analyst Capability	1.46	1.19	1.00	.86	.71	--	--
Applications Experience	1.29	1.13	1.00	.91	.82	--	--
Virtual Machine Experience	1.21	1.10	1.00	.90	.90	--	--
Language Experience	1.14	1.07	1.00	.95	.95	--	--
Execution Time	1.00	1.00	1.00	1.11	1.30	1.66	--
Main Storage Constraints	--	--	1.00	1.06	1.21	1.56	--
Virtual Machine Volatility	.87	.87	1.00	1.15	1.30	1.49	--
Computer Turnaround Time	.79	.87	1.00	1.07	1.15	--	--
Requirements Volatility	--	.91	1.00	1.19	1.38	1.62	--
Software Reliability	.75	.88	1.00	1.15	1.40	--	--
Data Base Size	--	.94	1.00	1.08	1.16	--	--
Software Product Complexity	.70	.85	1.00	1.15	1.30	1.65	--
Required Reuse	--	--	1.00	1.10	1.30	1.50	--
Modern Programming Prac.	1.24	1.10	1.00	.91	.82	--	--
Use of Software Tools	1.24	1.10	1.00	.91	.83	.73	.62
Management Reserve - Risk	1.00	1.20	1.40	1.60	1.80	2.00	--

	Unclassified	Classified
Classified Security Application	1.00	1.10

REVIC EFFORT AND SCHEDULE EQUATIONS

	EFFORT IN MAN MONTHS(MM)	SCHEDULE IN MONTHS
ADA	$(MM) = 6.8000\ (KDSI)^{0.94}$	$TDEV = 4.3760\ (MM)^{0.32}$
ORGANIC	$(MM) = 3.4644\ (KDSI)^{1.05}$	$TDEV = 3.6500\ (MM)^{0.38}$
SEMIDETACHED	$(MM) = 3.9700\ (KDSI)^{1.12}$	$TDEV = 3.8000\ (MM)^{0.35}$
EMBEDDED	$(MM) = 3.3120\ (KDSI)^{1.20}$	$TDEV = 4.3760\ (MM)^{0.32}$

REVAD DATA COLLECTION FORM

TACP CASE NUMBER ______________ PRICING CASE NUMBER ______________

DELIVERABLE SOURCE INSTRUCTIONS (DSI)
If more modules needed, continue on another form

NEW CODE	Low Estimate	Most Probable Estimate	High Estimate
Module 1	________	________	________
Module 2	________	________	________
Module 3	________	________	________
Module 4	________	________	________
Module 5	________	________	________

ADAPTED CODE	Adapted DSIs	% of Code to be Redesigned	% of Code to be Recoded	% of Integration Compared to Original
Module 1	________	____%	____%	____%
Module 2	________	____%	____%	____%
Module 3	________	____%	____%	____%
Module 4	________	____%	____%	____%
Module 5	________	____%	____%	____%

Weighted Dollars Per Manhour (Charged by Company) $________

Average Man Hours Per Man Month _______

Size of the Dynamic Memory of the Target Computer (in Bytes) ____________

Size of the Data Base (in Bytes or Words) _________

Programming Language Used to Develop Software ____________________

Developmental Team Language Experience (Avg. in Months) ____________

Systems Engineering Phase ________ %
REVIC defines the Systems Engineering Phase as spanning the time from contract award until the completion of the Software Specification Review. (SSR). REVIC calculates this effort as a percentage of the calculated development time. You may choose a different value from the default of 12%.

Development Test & Evaluation Phase ________ %
DT & E starts after the Formal Qualification Test at the CSCI level and continues until the completion of the Functional and Physical Configuration Audits at the System Level. The default value is 22 .

REVAD COMMENTS/QUESTIONS

Your comments and questions are valued and help ensure that both REVAD and REVIC are useful to you, the DLA Software Cost Estimator.

User Information

Office:
Estimator Name:
Address:
Commercial Phone Number: Autovon:

Proposal Information

Description of Software Product:

Development Mode: Number of Modules:

Number of Deliverable Source Instructions:

Programming Language(s):

Comments/Questions:

Please submit all comments/questions to DLA-EE at the following address:

HQ DLA
Attn: DLA-EE
Cameron Station
Alexandria, VA 22304-6100